Technology Consulting Company
Research, Development &
Global Standard

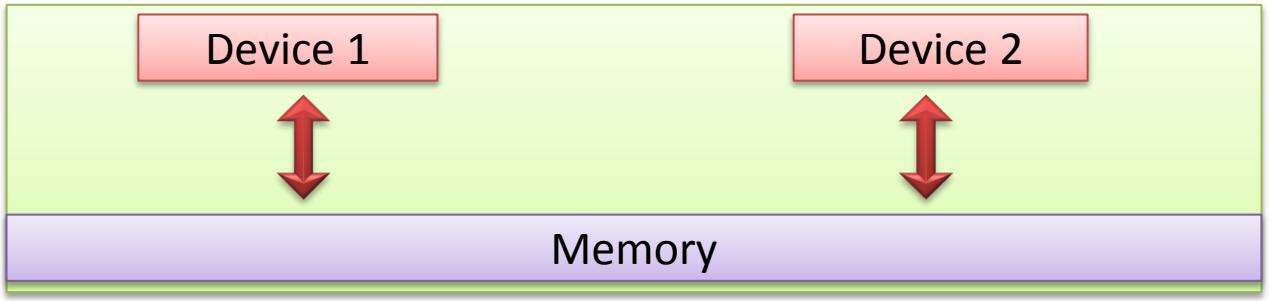
Linux Contiguous Memory Allocator (and a little IOMMU)

Damian Hobson-Garcia
Igel Co., Ltd.

DMA memory transfers

igel

- Modern SoC's can perform many operations with minimal CPU processing
  - Examples:
    - Video encode/decode
    - Image processing
    - Audio encode/decode
- Data processing usually involve some kind of DMA
  - device-to-device
  - device-to-memory



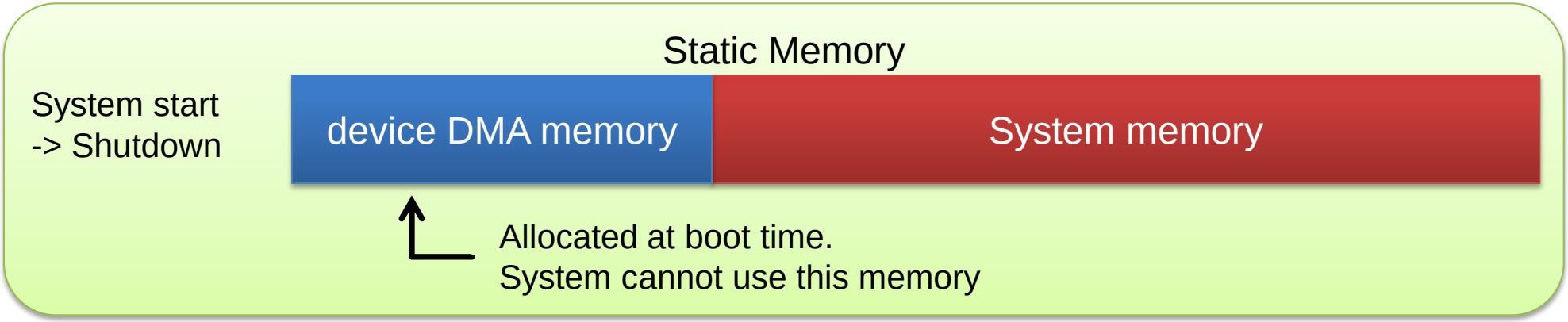
DMA-able memory

igel

- some devices can do scatter-gather memory access, but many cannot.
  - Maybe you have an IOMMU to do this for you (more later)
- Need physically contiguous memory for DMA
- Can do static or dynamic allocation

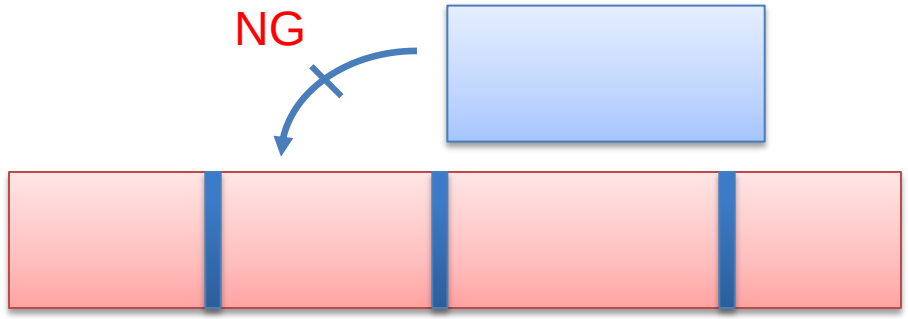
Static Allocation

- Allocate memory at boot time
- Simple to do (eg. via memblock)
- waste of memory when device is not in use

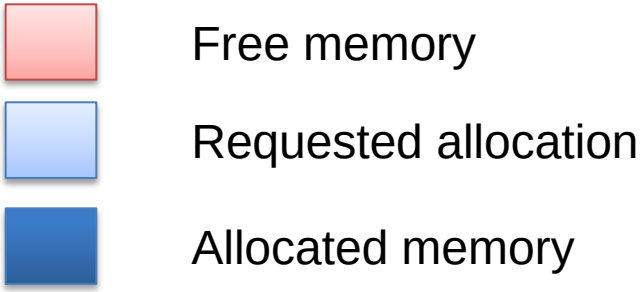


Dynamic Allocation

- Allocate when device is opened
- Uses memory more efficiently
- memory fragmentation problem
  （bigger memory allocations = bigger fragmentation problems)

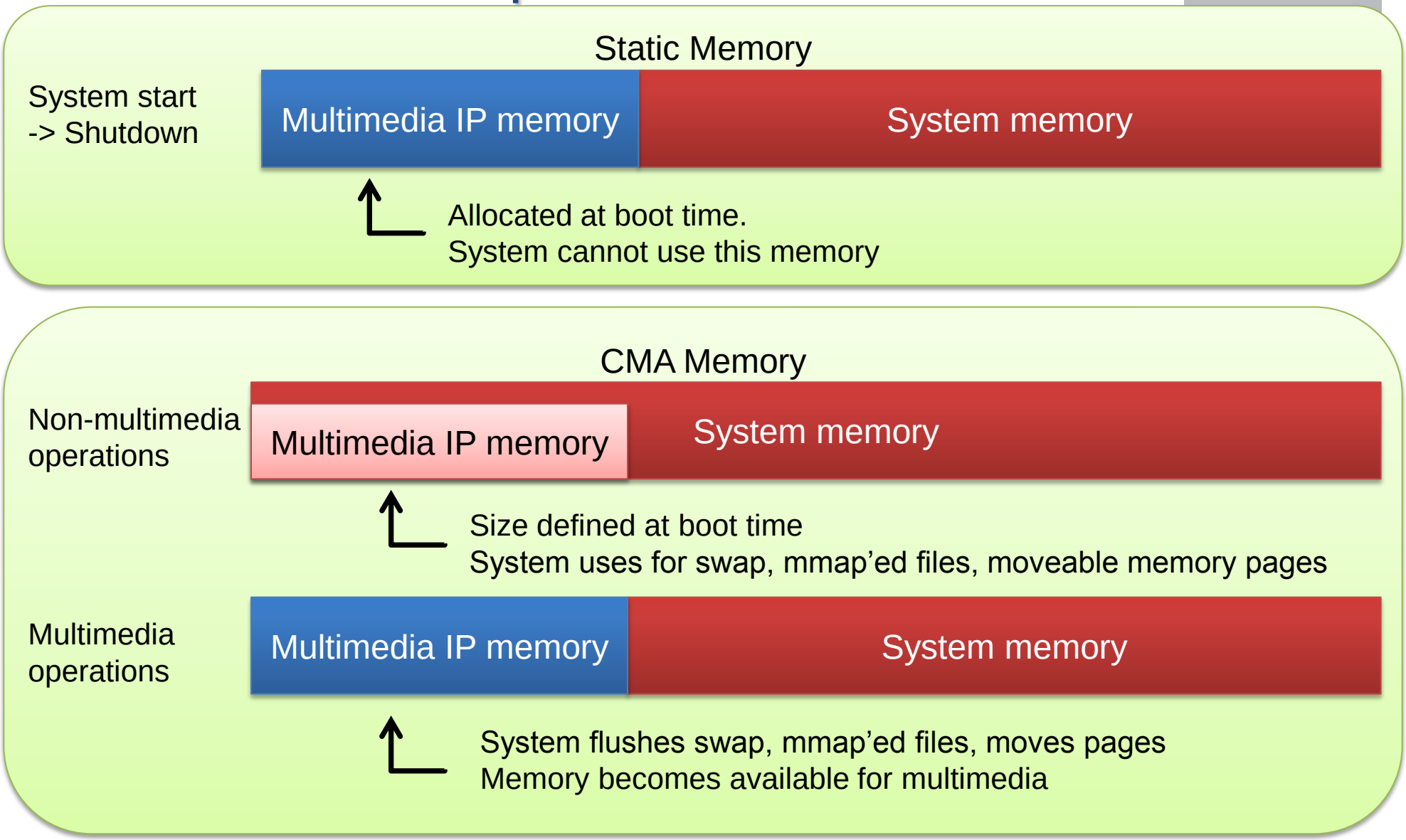


- Example
  - Free memory (>90%)
  - Requested allocation (~30%)
  - Result: FAIL



Contiguous Memory Allocator (CMA) can help solve the memory fragmentation problem

Static Memory vs. CMA memory
Multimedia example



Contiguous Memory Allocator

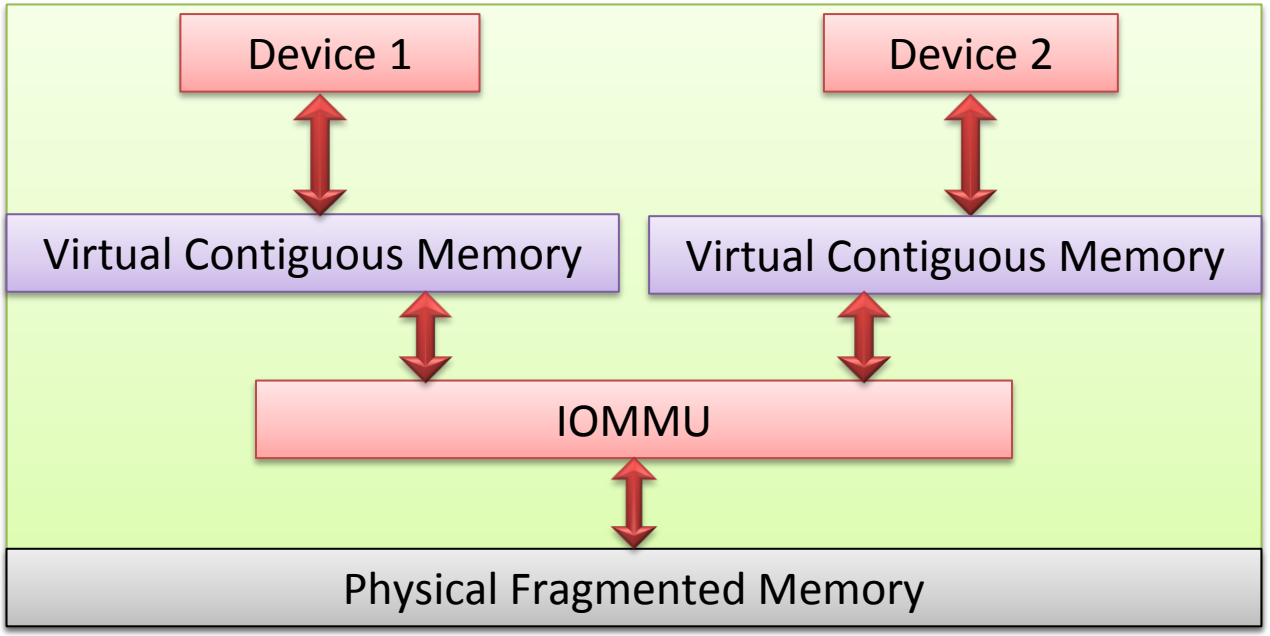
igel

- CMA support introduced in Linux kernel v3.5-rc1
  - Additional bug fixes throughout v3.5-rc, v3.6…
- Backported to LTSI kernel v3.4
- Global and per-device CMA pools
  - Good for devices with very large memory requirements
- Access via DMA mapping API
  - eg. dma_alloc_coherent()
  - many drivers already use this API, so no changes are necessary
  - UIO driver did NOT use this API, so new UIO device added

Non-contiguous memory via IOMMU

igel

- Some devices provide central MMU that can perform scatter-gather for other devices
- MMU for Input/Output => IOMMU
- Another way to solve dynamic DMA memory fragmentation

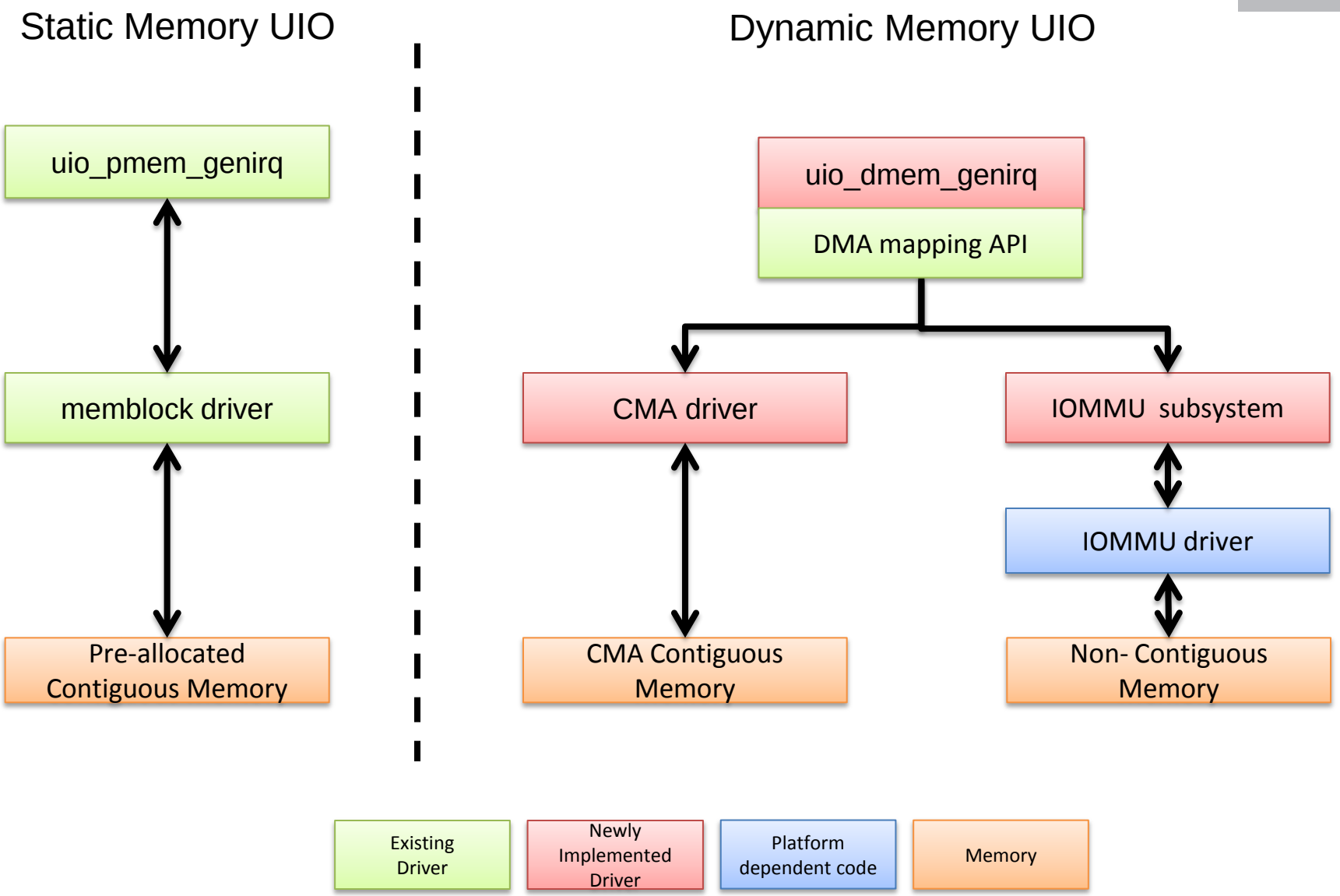


IOMMU support

igel

- IOMMU subsystem added together with CMA
  - Also backported to LTSI v3.4
- Need to have (write) IOMMU driver for your MMU hardware
  - see: drivers/iommu/* for examples
- IOMMU allocated memory available via DMA mapping API

Memory Allocation with UIO



Dynamic Memory UIO

igel

- New dynamic memory UIO platform driver (uio_dmem_genirq)
- Based on platform UIO device driver (uio_pdrv_genirq)
  - same irq handling and static memory region support
- Backward compatible with current platform user-space drivers

Dynamic Memory UIO usage

igel

- Define **size** and **number** of dynamic regions in platform data (max # of regions = 5 as per UIO spec)
- Opening /dev/uioX will allocate all dynamic regions for device
- Get memory addresses:
  - $ cat */sys/class/uio/uioX/maps/mapY/addr*
- Closing /dev/uioX frees the memory

Summary

igel

- CMA and IOMMU allow for dynamically allocating DMA memory
- Support added in v3.5, v3.6
  - Backported to LTSI v3.4
- Usable by kernel drivers via DMA mapping API
- Useable by user space drivers via uio_dmem_genirq driver